



**QUEEN'S
UNIVERSITY
BELFAST**

Automatic generation of multiblock decompositions of surfaces

Fogg, H. J., Armstrong, C. G., & Robinson, T. T. (2015). Automatic generation of multiblock decompositions of surfaces. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, 101(13), 695-991. <https://doi.org/10.1002/nme.4825>

Published in:
INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING

Document Version:
Peer reviewed version

Queen's University Belfast - Research Portal:
[Link to publication record in Queen's University Belfast Research Portal](#)

Publisher rights
Copyright © 2015 John Wiley & Sons, Ltd.

This is the accepted version of the following article: Fogg, HJ, Armstrong, CG & Robinson, TT 2015, 'Automatic generation of multiblock decompositions of surfaces' *International Journal for Numerical Methods in Engineering*, vol 101, no. 13, pp. 695-991, which has been published in final form at <http://onlinelibrary.wiley.com/doi/10.1002/nme.4825/abstract>

General rights
Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

Automatic generation of multiblock decompositions of surfaces[†]

Harold J. Fogg, Cecil G. Armstrong and Trevor T. Robinson*

School of Mechanical and Aerospace Engineering, Queen's University Belfast, BT9 5AH, N. Ireland

SUMMARY

Multiblock-structured meshes have significant advantages over fully unstructured meshes in numerical simulation but automatically generating these meshes is considerably more difficult. A method is described herein for automatically generating high-quality multiblock decompositions of surfaces with boundaries. Controllability and flexibility are useful capabilities of the method. Additional alignment constraints for forcing the appearance of particular features in the decomposition can be easily handled. Also, adjustments are made according to input metric tensor fields that describe target element size properties. The general solution strategy is based around using a four-way symmetry vector-field, called a cross-field, to describe the local mesh orientation on a triangulation of the surface. Initialisation is performed by propagating the boundary alignment constraints to the interior in a fast marching method. This is similar in a way to an advancing-front or paving method but much more straightforward and flexible because mesh connectivity does not have to be managed in the cross-field. Multiblock decompositions are generated by tracing the separatrices of the cross-field to partition the surface into quadrilateral blocks with square corners. The final task of meshing the decomposition requires solving an integer programming problem for block division numbers. Copyright © 2010 John Wiley & Sons, Ltd.

Received ...

KEY WORDS: multiblock decomposition; block-structured mesh; mesh singularities; cross-field

1. INTRODUCTION

Structured grids with regular (i, j) connectivity have many advantages for numerical simulation; they produce sparse banded system matrices, they are ideal for multigrid acceleration and they facilitate anisotropic stretching without severe degradation of the element shape qualities to name a few. However, wrapping a structured grid around a complex geometry while maintaining boundary conformity can be a tough problem and will usually cause high levels of undesirable distortion. The rigid connectivity is too restrictive in most cases and a few well-placed nodes where the grid structure is disrupted, called *singularities*, can dramatically reduce the overall distortion of the mesh and improve its usefulness. This leads to the idea of a *multiblock-structured* or sometimes just *block-structured* mesh where the simulation geometry is divided into four-sided subregions called blocks, each of which is filled with a regular grid. A multiblock decomposition is essentially an extremely coarse mesh where each block is empty and it acts as the framework for defining the shape and topology of a block-structured mesh. Mesh singularities occur at junctions where there are more or less than four neighbouring blocks.

Block-structured meshes are often favoured by analysts and they are sometimes required for particular numerical schemes. They are used if at all possible in dynamic structural mechanics analyses with explicit solvers [1, 2] and are almost essential for the Spectral Element Method [3, 4].

*Correspondence to: Trevor T. Robinson, School of Mechanical and Aerospace Engineering, Queen's University Belfast, BT9 5AH, N. Ireland, t.t.robinson@qub.ac.uk.

[†]The work was sponsored by the Aircraft Research Association (ARA).

The extremely anisotropic features that regularly appear in CFD analyses, such as boundary layers, wakes and shock fronts, mean that there is a major incentive to use locally regular grids. The advantages of using locally structured grids in CFD are widely accepted and reported in [5, 6, 7]. Ideally, the block-structured meshes that are used would have shapes and topologies that are specifically geared towards efficiently capturing the solution characteristics.

Generating such meshes remains to be a challenge, most especially in 3-D for hexahedral meshes. In 2-D, a few methods are established for creating quadrilateral (quad) meshes such as paving [8, 9], Cartesian grid [10], submapping [11], and medial-axis based decomposition methods [12, 13] but they all fall short in one or more aspects. Paving often creates a poor mesh topology with numerous unnecessary mesh singularities. Cartesian grid methods produce unnatural mesh topologies with poor element qualities on boundaries. Submapping methods are only effective on a small class of geometry. Medial axis decomposition methods are perhaps the most capable algorithms. They can often generate block-structured meshes of reasonably good quality. However, medial axis degeneracies and concave corners are not always dealt with in the best way and there is little scope for accommodating target mesh properties. Manually building a block-structured mesh is still common practice because meshes of higher quality can be generated and their shape and topology can be carefully controlled. But this is a time consuming process and requires significant skill and experience on the part of the user.

2. RELATED WORK

2.1. Cross-fields for mesh generation

There has been a marked trend toward cross-field guided mesh generation in recent years. A cross-field is a four-way rotationally symmetric field where each point has an assigned set of four unit vectors on the tangent plane which form a regular cross. They first appeared in computer graphics applications as a useful means to control a mapping to a surface for non-photorealistic rendering [14], texture synthesis [15] and remeshing and global parameterisation [16]. The theory of N-symmetry fields and their singularities were explored by Palacios [17] and Ray [18]. An equivalent theory was arrived at from a different point of view by Bunin [19] by studying the properties of a conformal mapping from a notional flat surface with cone points (where the curvature is concentrated at a point) with fixed angle defects to an arbitrary surface. The behaviour of a N-symmetry field and the variation in element size of a corresponding mesh are shown to be inextricably linked and they are both essentially governed by the Poisson equation with point source terms at the cone points, which is where singularities occur. This theory demonstrates that the problems of designing cross-fields and meshes are in the class of Inverse Poisson problems, characterised by some information of a potential field from which a suitable charge distribution must be reconstructed. These are known to be ill-posed problems that are difficult to solve effectively [20].

2.1.1. Cross-field generation. For curved surfaces, it is well known that the error of piecewise linear approximations is minimised by aligning elements with the principal curvature directions [21]. Moreover, with hex mesh generation in mind, Dupin's Theorem [22] of triply orthogonal systems of surfaces essentially states that hex elements may only preserve orthogonal corner angles if constituent sheets of quad elements have edges that are principal curves of those sheets. Therefore, it is favourable for cross-fields to be aligned with the principle curvature directions of the surface. The strategy of generating cross-fields by approximating the principal curvature directions have been used in [16, 23, 24]. This approach has the advantage of producing appealing cross-fields that are coupled to inherent surface properties by a straightforward routine and singularities appear naturally at umbilic points. The main disadvantage is that in large regions where the normal curvature is close to being symmetric (*i.e.* similar principal curvature values, κ_1 and κ_2), which are not uncommon, the cross-field is poorly defined. Examples of surfaces with anisotropic and isotropic normal curvatures are shown in Fig. 1. Also, the alignment of the cross-field with the surface boundaries is not handled naturally.

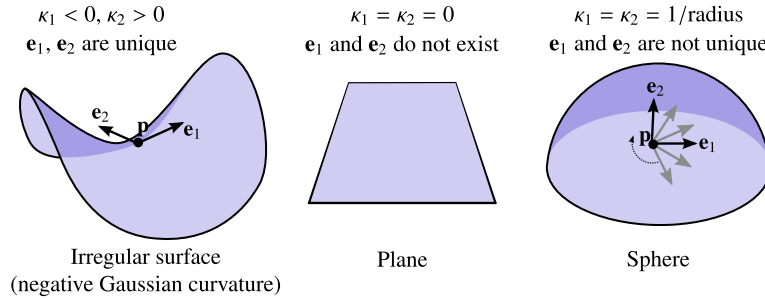


Figure 1. An example of a surface with anisotropic normal curvature and unique principal directions (e_1, e_2) (left) and two examples of surfaces with isotropic normal curvatures and no unique principal directions (centre and right).

For surfaces with regions of symmetric curvature, large total Gaussian curvature indicates that a singularities should occur in the locality. Bunin's theory [19] (briefly summarised in Sec. 3) substantiates this claim. For example, a plane has a uniform Gaussian curvature of zero, hence, it requires no singularities (in the absence of alignment constraints). Closed surfaces have a total Gaussian curvature value that is solely dependent on its genus which puts a limit on the minimum number of singularities necessary to mesh it. In the setting of conformal parametrisation, Ben-Chen *et al.* [25] devised a method for identifying suitable cone singularities by placing them in the centres of regions with significant total Gaussian curvature in a 'greedy' optimisation method. However, the method is not designed specifically for cross-fields with fixed singularity types.

For cross-fields on 2-D planes, singularities are required either to satisfy boundary alignment constraints or to facilitate quad element size and direction targets. Bunin [26] describes a method to directly tackle the Inverse Poisson problem on flat surfaces. As inputs, Neumann (for directions) and Dirichlet (for isotropic sizes) boundary conditions are assigned and a point source distribution representing mesh singularities is solved which satisfies these to some extent. The method is appealing from a theoretical point of view and some impressive results are shown. However, it appears to contain some steps which require careful fine-tuning and high sensitivities to numerical error are reported for problems with large numbers of singularities.

A common starting point for generating cross-fields on flat and curved surfaces is a sparse set of feature edges either taken from edges of a B-rep CAD geometry, numerically approximated by methods such as [27, 28] or assigned as control strokes by the user. With these acting as 'boundary conditions' there are a variety of techniques that have been devised to find a cross-field adhering to them. Bommes *et al.* [29] describes a mixed-integer linear optimisation formulation to find a smooth cross-field with singularities (with corresponding integer conditions) on a tri mesh. To reduce the search space, nearest-patches to assigned directions are obtained by a modified Dijkstra algorithm and singularities are forced to occur on patch boundaries. Their mixed-integer solver followed by a local singularity optimisation search produces good quality cross-fields, though the complexity of the algorithm is high and the performance is dependent on the construction of the Dijkstra trees to a large extent, of which minimal details are included. Kowalski *et al.* [30] use a (unit) vector representation of a cross-field that was originally described by Palacios *et al.* [17]. The representative vector field on a tri mesh is solved in two stages. First, the unit norm constraint is relaxed and a stationary heat equation with Dirichlet boundary conditions is solved in a FEM analysis. A following iterative optimisation method procures a valid unit vector field and determines the singularities. Drawbacks are that the method cannot easily handle curved surfaces and it is unsuited to adjusting for target element sizes and directions.

For optimising the smoothness of a cross-field a non-linear objective function was developed by Hertzmann [14] and used by Liu *et al.* [31] involving periodic trigonometric functions to give a modulo $\pi/2$ treatment of orientations. Quasi-Newton methods can be used to solve for a cross-field that minimises this function but an effective initialisation is essential because the occurrence of singularities causes non-convexities and random initial fields will inevitably lead to large numbers

of unnecessary singularities. Liu *et al.* [31] use a “light-weight” method for initialisation by extrapolating information from the alignment constraints. A queue of elements is iterated through until it is empty; if an element has enough neighbours with assigned crosses a new element is computed and removed from the queue. There is significant scope to improve on this method by making changes to ensure that the constraints have a fairly apportioned region of influence when the underlying tri mesh is of varying size. Additionally, making adjustments for accommodating target element sizes is also possible. These gaps are filled by the novel initialisation method presented in this paper.

2.1.2. Mesh generation using a guiding cross-field. The QuadCover algorithm [24] produces a globally continuous near orthogonal parametrisation except at singularities from an input cross-field where integer isoparametric curves describe a quadrilateral mesh. With the singularities explicitly defined, the square of the differences between the cross-field and parametric gradients are minimised. By relaxing the problem to ignore the integer constraints an approximate solution is found by linear algebra and a feasible solution is then sought by rounding to the nearest integers. The method will most often be capable generating valid quad meshes of small element sizes. However, the rounding tactic is too simple to be able to effectively deal with coarse meshes sizes. An iterative rounding scheme is used by Liu *et al.* [31] that may cope better with coarser sizes. Bommes *et al.* [29, 32] have devised sophisticated methods to find integer grid maps for generating quad meshes from cross-fields with sensitivities to size inputs. They combine greedy strategies for solving mixed integer programming problems and tri decimation methods to reduce its complexity.

Apart from parametrisation another method for generating a mesh is by creating a multiblock decomposition that is fitted to the cross-field. This automates the decomposition stage in the conventional multiblock mesh generation process. Regular regions of a cross-field not containing singularities can be mapped to a grid, hence, the cross-field streamlines connected to singularities have the effect of partitioning the surface into blocks where grids can be generated. Together these form what has been called the base complex [33], the quad patch complex [34] and the separatrix graph [17, 30] by different authors. For a surface with boundaries the streamlines connected to corners should also be included. Generating a mesh by first decomposing the surface by the separatrix graph was proposed by Kowalski *et al.* [30]. To generate a mesh on a multiblock decomposition integer division numbers that satisfy template constraints for the blocks must first be solved. Although methods have been developed for this task [35, 36, 37, 38] some improved functionality is still be needed as will be explained. A major problem with this approach is that separatrices may spiral indefinitely about holes. Another is that they can form long thin blocks with heights much smaller than the target element size. These problems are looked at in this paper and a solution by using multiblock decompositions with partial contiguity in block interfaces is proposed.

3. GOVERNING EQUATIONS OF CROSS-FIELDS FOR MESH GENERATION

A continuum theory of unstructured mesh generation was described by Bunin [19]. It precisely outlines the interplay between cross-field curvature and mesh size variation, and it sheds light on the fundamental importance of mesh singularities. The key results of Bunin’s paper are:

- The conformal mapping between two surfaces, S and \tilde{S} , the latter being locally flat with some discrete cone points, is governed by the Poisson equation,

$$\Delta_S \phi = K + \sum_{i=1}^N k_i \frac{\pi}{2} \delta_{\mathbf{p}_i}, \quad k_i \in \mathbb{Z} \geq -4. \quad (1)$$

Δ_S is the Laplace-Beltrami operator, K is the Gaussian curvature of S and the $\delta_{\mathbf{p}_i}$ terms are weighted Dirac delta functions describing discrete cone points of \tilde{S} with discrete total curvatures of $k_i \pi/2$ occurring at the cone vertices, \mathbf{p}_i . The cone points correspond to mesh singularities with their characteristic types given by the integers k_i . The integer value can be loosely interpreted as

the number of additional quad elements above four at a node, or more reliably as the number of corners more than four that the polygon mesh pattern around a node has. Examples of $-\phi$ -fields with different types of singularities are shown in Fig. 2. (A negative field is used because it

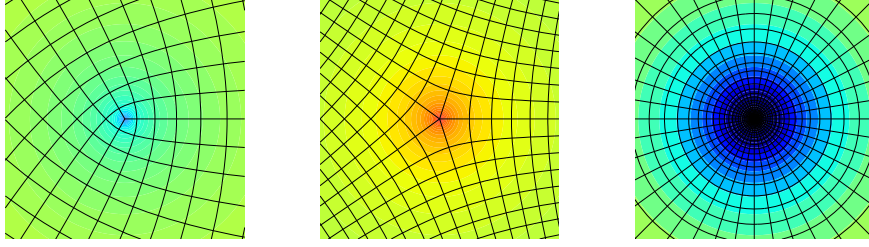


Figure 2. Mesh singularities and corresponding $-\phi$ -fields. Using polar coordinates centred at the singularity, ϕ is described locally by $k \frac{\pi/2}{2\pi} \ln(r)$. (Left) $k = -1$, a 3 valent node called a *negative singularity*; (centre) $k = +1$, a 5 valent node called a *positive singularity*; (right) $k = -4$, a polar grid around a higher-order singularity produced by super-imposing four negative singularities. (Taken and adapted from [19])

corresponds to temperature in an analogous stationary heat conduction analysis.)

- If a unit grid is drawn on \tilde{S} , its image on S is a locally isotropic grid of size

$$h = e^{-\phi}. \quad (2)$$

- The curvatures of the grid edges on S are related to the variation in the ϕ -field by

$$\kappa_g = \frac{\partial \phi}{\partial e} \equiv \langle \nabla_S \phi, \mathbf{e} \rangle, \quad (3)$$

where κ_g denotes the geodesic curvature of the edge. The intrinsic normal, \mathbf{e} , is defined by $\mathbf{e} = \mathbf{n} \times \mathbf{t}$ where \mathbf{n} is the surface normal and \mathbf{t} is the tangent vector of the edge on the surface. Thus, $\partial/\partial e$ represents the directional derivative along the intrinsic normal. This is illustrated

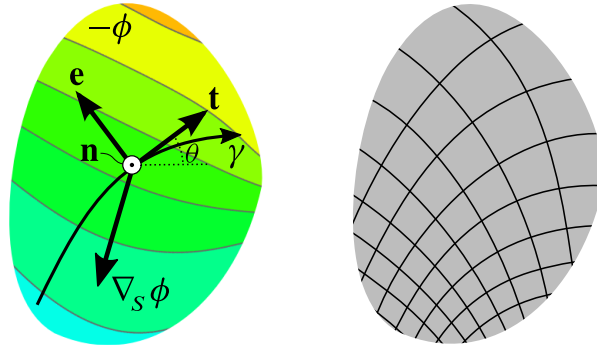


Figure 3. A mesh edge, γ , curved according to the local $-\phi$ -field (left) and a sketch of a corresponding isotropic mesh (right).

in Fig. 3. Angular changes of a cross-field between the points \mathbf{p}_1 and \mathbf{p}_2 on a curved surface are properly measured by

$$\Delta\theta = \angle (PT_{\mathbf{p}_1 \rightarrow \mathbf{p}_2} \mathbf{c}(\mathbf{p}_1), \mathbf{c}(\mathbf{p}_2)), \quad (4)$$

where PT is the parallel-transport operator and \mathbf{c} represents one of the four cross vectors. Then Eqn. (3) gives the differential expression

$$\kappa_g \equiv \frac{\partial \theta}{\partial t} = \frac{\partial \phi}{\partial e}, \quad (5)$$

which relates the ϕ -field and the relative angles of crosses in a cross-field. (Note that the directional derivatives are as defined in Eqn. (3)).

The ϕ -field acts as a continuum description of an infinitesimal isotropic quad mesh, or alternatively, a cross-field. Once a valid configuration of mesh singularities has been established standard finite element methods can be used to find a solution with Neumann boundary conditions applied to enforce boundary alignment constraints.

4. GENERATION OF A CROSS-FIELD

The main idea presented here is to initialise a cross-field over a whole surface from prescribed crosses on boundaries by propagating the constraints into the interior by a Fast Marching Method [39]. The strategy is carried out efficiently on an unstructured tri mesh and it produces an orderly cross-field with singularities pushed to the medial axis of the surface. New crosses are computed at the propagation front by making a compromise between maximising the smoothness of the cross-field and increasing its compatibility with a target size field that describes the ideal sizes of the mesh elements. A small amount of smoothing is required to improve the cross-field across the medial axis and to make fine adjustments. The method is most suitable for planar domains or spherical patches with numerous boundary features although it can applied to general curved surfaces also.

4.1. Target mesh properties

A metric space defined on the surface can effectively encapsulate the ideal size properties of a mesh. Complete mathematical descriptions of their theory can be found in [40] and more accessible summaries for mesh adaptation in [41, 42]. Practically, it is represented by a field of metric tensors that redefine how distances and angles are locally measured. The length of a tangent vector in metric space (in the basis on the tangent plane associated with the parametrisation) is computed by

$$\|\mathbf{u}\|_{\mathbf{M}} = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle_{\mathbf{M}}} = \sqrt{\mathbf{u}^T \mathbf{M} \mathbf{u}} \quad (6)$$

and relative angles by

$$\cos \vartheta_{\mathbf{M}} = \frac{\langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{M}}}{\|\mathbf{u}\|_{\mathbf{M}} \|\mathbf{v}\|_{\mathbf{M}}}, \quad (7)$$

where \mathbf{M} is the local metric tensor, a 2×2 symmetric positive-definite matrix. The unit ball in metric space is given by

$$B_{\mathbf{M}}(1) = \{\mathbf{x} \mid \|\mathbf{x}\|_{\mathbf{M}}^2 \leq 1\} = \{\mathbf{x} \mid \mathbf{x}^T \mathbf{M} \mathbf{x} \leq 1\}, \quad (8)$$

where $\mathbf{x} = (x, y)$ is the displacement vector on the tangent plane. Using an isothermal coordinate system (which can always be found), a coordinate change on the tangent plane to an orthonormal basis with eigen-vectors of \mathbf{M} , \mathbf{e}_1 and \mathbf{e}_2 , is performed by $\mathbf{x} \rightarrow [\mathbf{e}_1 \mathbf{e}_2]^T \mathbf{x}$. Thence, the equation of an ellipse become apparent,

$$B_{\mathbf{M}}(1) = \{\mathbf{x} \mid (x/h_1)^2 + (y/h_2)^2 \leq 1\}, \quad (9)$$

where h_1 and h_2 are the inverse square roots of the eigen-values of \mathbf{M} . The mapping of the unit ball in Euclidean metric space to the physical space is illustrated in Fig. 4 (left). A metric tensor implies two desirable properties of a quad element. First, that its edges have unit length when measured in metric space. And secondly, that its edges are aligned with the eigen-vectors so as to maintain perpendicular corner angles. The preservation of perpendicular corners and skewing of quasi-unit elements which are respectively aligned and misaligned with the eigen-vectors of \mathbf{M} is demonstrated in Fig. 4 (right).

Appropriate metric tensor fields for controlling mesh sizing can be obtained *a posteriori* from approximate simulation solutions [21, 42]. Since solution information is not generally available, *a priori* techniques for constructing appropriate metric tensor fields based on geometrical features have also been developed [43, 44, 45]. Artificial metric fields are used in the present work that are

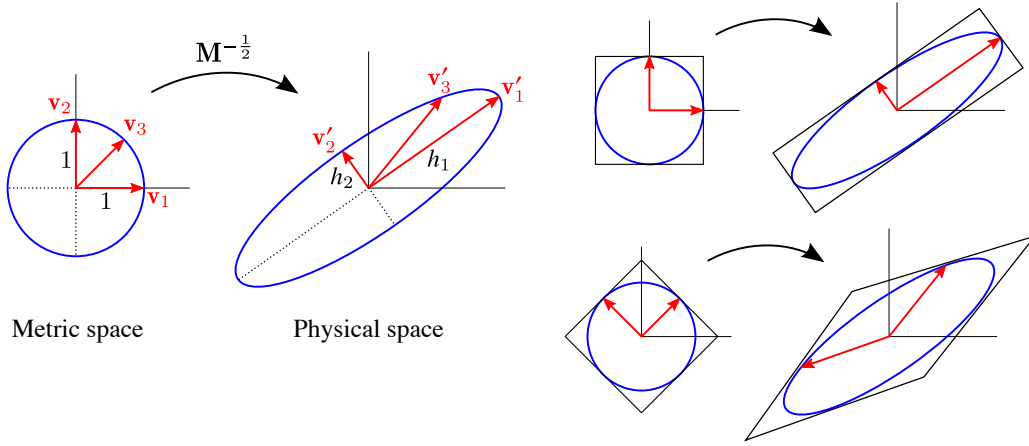


Figure 4. Mapping from Euclidean metric space to physical space (based on than in [42]) (left) and a demonstration of the dependency of orientation of quasi-unit elements on skewness (right).

defined by analytic functions or using a tool in CADfix [46] that interpolates a metric tensor field from specified metric tensors at points.

Equations (2) and (5) imply that variation in element sizes causes variation in cross angles. In a tri element with isotropic metric tensors defined at the three nodes, the variation in their size, \bar{h} , implies an optimum change in orientation of crosses over the tri element. (An overbar is used to signify a target property associated with the metric space.) Values for $\bar{\phi}$ can be computed by $-\ln \bar{h}$ (Eqn. 2). Approximating the surface locally as the planar tri element and assuming linear variations in parameters, a constant optimum gradient, $\bar{\nabla}\phi$, can be computed. From Eqn. (5), the accompanying target gradient in cross angle, $\bar{\nabla}\theta$, has the same magnitude and differs in orientation by $-\pi/2$ about the surface normal. A sketch of the target gradients that are computed from the metric tensors of the nodes of a tri element is shown in Fig. 5 (left). The optimum curvature of a mesh edge (dotted red line) with tangent \mathbf{t} is also illustrated. When non-isotropic metric tensors are involved the averages of the eigen-values are used to compute the target gradients.

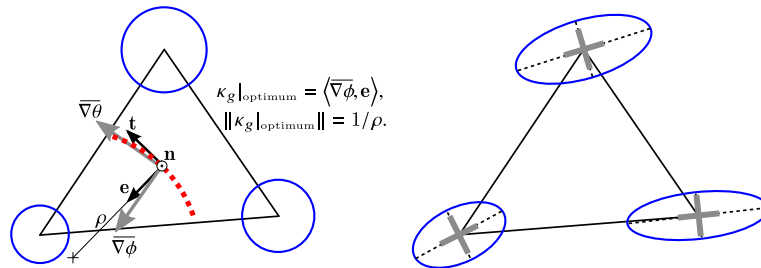


Figure 5. Target cross-field properties indicated by a metric tensor field. Variation in isotropic size gives a target gradient of cross angle (left) and anisotropy gives target cross orientations (right).

Anisotropic metric tensors describe a preferred orientation of quad elements with edges aligned with their eigen-vectors. The degree of importance of the target orientation depends on the anisotropy of the principle sizes, i.e. $\max(h_1, h_2)/\min(h_1, h_2)$. Examples of target cross orientations associated with anisotropic metric tensors are shown in Fig. 5 (right) with the importance of the bottom right being greatest and the bottom left being least.

4.2. Boundary alignment constraints

Boundary conformity is a typical requirement of the mesh. Thus, the sought cross-field should have streamlines through the boundary edges. In other words, the two orthogonal *directions* (i.e. one dimensional subspaces) associated with a cross on a boundary edge are along and perpendicular to

the edge tangent. However, a cross cannot simultaneously have directions along both tangent vectors of the connected edges of a corner in general. The best fit cross has a direction that is either aligned with the corner bisector or offset by $\pi/4$. The integer

$$n_c = \text{round}\left(\frac{\theta_c}{\pi/2}\right) \quad (10)$$

indicates the optimum cross alignment at a corner with angle θ_c . The parity of n_c indicates the optimum alignment of a cross with the corner bisector: even – aligned, odd – offset. Intuitively, n_c describes the number of elements to occur at the corner.

4.3. Penalty energy functional

A method for measuring the local quality of a cross-field is useful. All things being equal, rapid variation in cross angles is undesirable as it reduces the cross-field smoothness. The Dirichlet energy of Bunin's ϕ -field is a natural way to assess the smoothness,

$$\begin{aligned} E_{\text{smoo}} &= \int \|\nabla\phi\|^2 dA \\ &= \int \|\nabla\theta\|^2 dA \quad (\text{if planar}). \end{aligned} \quad (11)$$

Deviation in element size variation from target size variation can be measured by

$$E_{\text{grad}} = \int \|\nabla\theta - \bar{\nabla}\theta\|^2 dA. \quad (12)$$

And similarly for deviation of crosses from target directions,

$$E_{\text{direc}} = \frac{1}{A_{\text{total}}} \int (\theta - \bar{\theta})^2 dA, \quad (13)$$

where the total area divisor has been included so that the expression yields a dimensionless value. The overall energy functional is a combination of the three individual terms

$$E = E_{\text{smoo}} + w_1 E_{\text{grad}} + w_2 E_{\text{direc}}, \quad (14)$$

with weights, $w_1, w_2 \in \mathbb{R} > 0$, assigned so as to capture how crucial or not the target size gradients and directions are. For example, w_1 depends on how vital the assigned size variation is and w_2 should be proportional to the anisotropy of the metric tensor field.

A simple formula is used during the initialisation method to solve for a locally optimum cross orientation at a free node in a tri element. Piecewise linearity of θ , ϕ and $\bar{\theta}$ is assumed. The formula is derived by setting the numerical derivative of Eqn. (14) equal to zero. The details are given in App. A.

4.4. Overview of algorithm

The proposed method for the generation of a cross-field consists of the high-level steps itemised in Alg. 1. Illustrations of the different steps are shown in Fig. 6.

The method involves finding the optimal cross at one node and then the next with consideration to the local cross-field. By visiting nodes in order of increasing distance from the boundaries the solution varies smoothly except along the medial axis. Singularities usually occur on the medial axis except in unusual cases when high priority grading and anisotropy dictate otherwise.

4.5. Appropriate tri mesh

There are a number of things to consider when selecting the sizing of the tri mesh used by the method. The cross-field is treated as varying linearly over tri elements so the sizing of the tri mesh

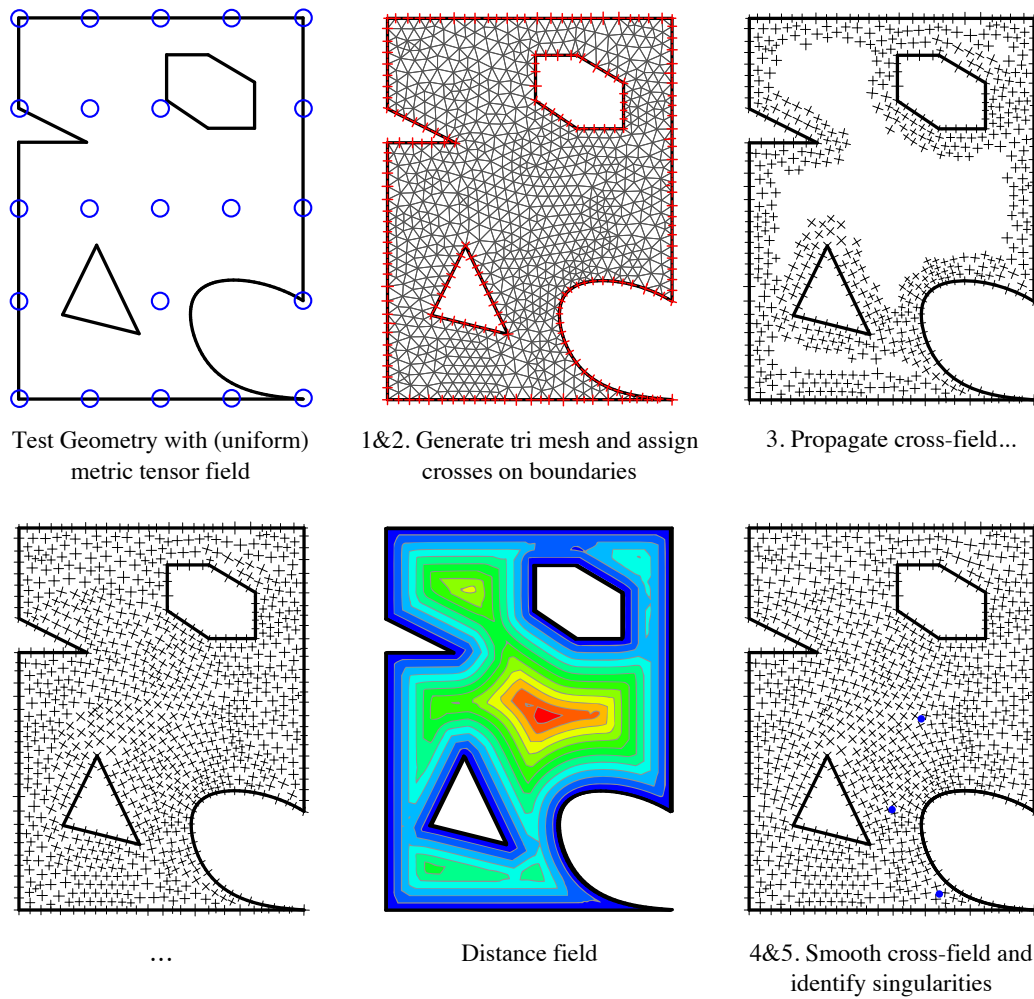


Figure 6. Cross-field generation process.

Algorithm 1 Generate cross-field

1. Generate a tri mesh on the surface.
2. Assign crosses on boundaries.
3. Propagate the cross-field from the boundaries to the interior. New crosses are solved at the propagation front by minimising Eqn. (14). The method is organised in the manner of the *Fast Marching Method*.
4. Apply some smoothing to the cross-field.
5. Identify singularities in tri elements by a simple check.

should be reasonably small in order captured its behaviour. On the other hand, there is an incentive to limit the number of nodes and elements for efficiency. Although the specific details of the solution are unknown to start with, node spacing on edges in proportion to the edge curvature is always appropriate. Also, the tri mesh should be a fitting representation of the surface with element sizes small enough to reproduce small features. If it is wished for small geometric features of the surface be absent from the quad mesh then they can be simply 'meshed over' by the tri mesh. Since the input metric tensor field instructs how the quad mesh would ideally behave it is prudent to use a tri mesh

capable of capturing that behaviour. Finally, for curved surfaces the normal curvature puts a limit on the maximum size that is appropriate for realising an adequate piecewise linear approximation of the surface.

Whilst the performance of the method is not overly dependent on the quality of the tri mesh, and it will still produce reasonable solutions with poor tri meshes, good performance is ensured by taking some care to see that the tri mesh elements have reasonable angles and smoothly varying sizes. Given the wide availability of effective tri mesh generation and improvement methods there is no good reason to use poor quality tri meshes.

4.6. Initialisation of the cross-field by propagation

The standard Eikonal equation is solved to give a distance field with values of zero at the boundaries. Its purpose is to guide the order in which new crosses are computed to fairly apportion the region of influence of each boundary and promote regularity. A version of the Fast Marching method is used for this which is based on the finite element approach described in [47].

Pseudo-code of the algorithm is given in Alg. 2. The hyperbolic property of the Eikonal equation where the solution at a point is only affected by the solution ‘upwind’ of that point is taken advantage of. The nodes are labelled either *alive* (computed and fixed assigned distance), *narrow-band* (just computed distance) or *far-away* (not touched yet). As yet unassigned distances can be computed at nodes by extrapolation over tri elements with two nodes with assigned distances and marked as *alive*. Assumed local linear variation in the distance field and a gradient of unit norm is enough to make this calculation. A node with a newly computed distance is re-marked as *narrow-band*. The method proceeds by iteratively re-marking the *narrow-band* node with the smallest distance as *alive* and then checking if the distances of its *far-away* neighbours can be computed. Nodes at which new distances are solved are also assigned a cross by locally minimising Eqn. 14.

Algorithm 2 Cross-field propagation

```

▷ Initialisation:
narrowBandList ← []
for every node  $i$  do
  if node  $i$  on boundary then
    mark node  $i$  as alive
     $cross_i$  ← boundary aligned cross
     $d_i$  ← 0
  else
    mark node as far-away
for neighbour elements of all alive nodes do
  —Repeated Block—
  if neighbour element has two alive nodes (1,2) and one far-away or narrowband (3) then
     $d_3^* \leftarrow$  compute  $d \dots$  (linear interpolation of  $d$ ,  $\nabla d = 1$ )
    if  $d_3 = \text{None}$  or  $d_3^* < d_3$  then
       $d_3 \leftarrow d_3^*$ 
       $cross_3 \leftarrow$  compute  $cross \dots$  (by minimising Eqn. (14))
      mark node 3 as narrowband
       $\text{heapq.push}(\text{narrowbandList}, (d_3, \text{node } 3))$ 



---


▷ Loop:
while narrowbandList  $\neq []$  do
  ( $d_{\text{smallest}}, trial$ ) ←  $\text{heapq.pop}(\text{narrowbandList})$ 
  mark  $trial$  as alive
  for neighbour elements of  $trial$  do
    —Repeated Block—
    ...

```

A *min-heap priority queue* is used to efficiently find the minimum distance node in the *narrowbandList*. The computational complexity of the procedure is $\mathcal{O}(N \log N)$. The *heapq* module in Python [48] is used in the implementation.

Results of the propagation method are shown in Fig. 7 for the simple example where two crosses have been assigned on a plane. A regular tri mesh with nodes at the assigned crosses is used. The

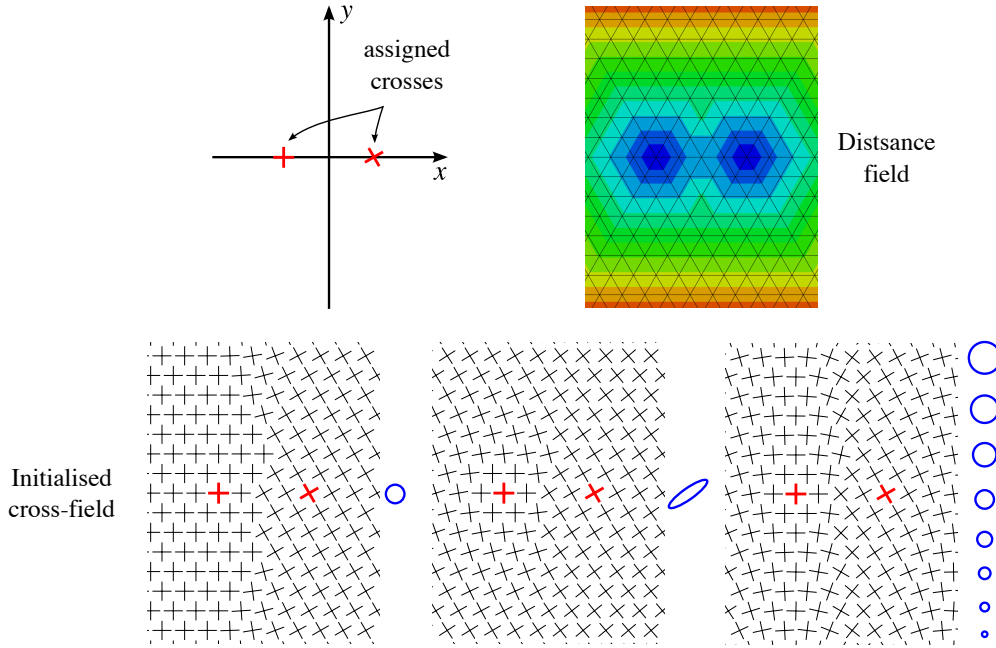


Figure 7. Demonstration of cross-field initialisation method for assigned crosses at two points on a plane with three different input metric tensor fields – isotropic uniform, anisotropic uniform and isotropic varying.

imprint of the mesh appears in the distance field solution because the simple piecewise linear approximation is ineffective at capturing the high curvatures of distance contours near the two assigned points. Results are shown for three different metric tensor fields, as indicated by the blue ellipses. Perceptible in each of the three results is the characteristic disjointedness of the cross-fields across the medial axis, which in this case is the y -axis.

4.7. Smoothing of the cross-field

A smoothing method is used to remove the sharp discontinuities across the medial axis resulting from the initialisation method. The standard way to formulate a smoothing method is to define a measure of the distortion of the field as an objective function and then employ an optimisation method to minimise it in order to create the improved field. The Dirichlet energy (Eqn. 11) springs to mind as a suitable (quadratic) objective function but singularities need to be defined or else integer variables must be used to handle their occurrences. A more appealing option is to use a non-polynomial objective function which is only sensitive to modulo $\pi/2$ differences in angle of crosses by using quadruple-angle trigonometric functions. This way, it is immaterial whether or not a singularity occurs in a tri element with respect to the distortion value. However, it is obviously periodic with major non-convexity thus gradient based optimisation can only hope to find locally optimum solutions which are highly dependent on the initial cross-field.

The smoothness function introduced by Hertzmann *et al.* [14] to measure the misalignment between two crosses is

$$\xi(\theta_1, \theta_2) = \frac{1}{2}(1 - \cos(4\theta_1 - 4\theta_2)), \quad (15)$$

where θ_1 and θ_2 are the angles between any of the four cross directions and a local reference axis. The function is plotted in Fig. 8. An approximation of the total distortion in over a tri element, e ,

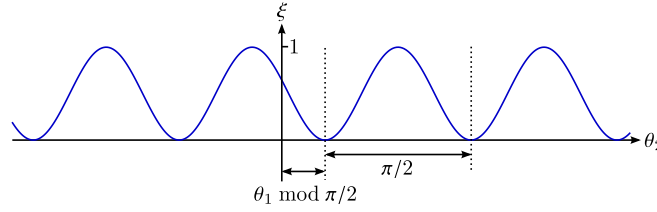


Figure 8. Periodic smoothness function between two crosses.

with θ_1 as a reference is

$$\Xi_e = \frac{1}{A_{\text{total}}} \iint_e \xi^{(e)}(\theta_1, \theta(\mathbf{x})) dA, \quad (16)$$

where a simple (e.g. linear) polynomial is used to interpolate ξ over the element. The total distortion for the whole cross-field is the sum over each tri element,

$$\Xi = \sum_e \Xi_e. \quad (17)$$

The minimisation of this as the objective function, with an array of solution variables of cross angles for each node, is a large and numerically demanding problem. To simplify the problem it is separated into a series of reduced problems that only involve the most dominant solution variables. This produces a more manageable problem that can be solved more economically. Hence, the problem of minimising Eqn. (17) considering all N nodes in the tri mesh is broken down into N problems of minimising each node angle considering only the nodes in adjacent elements. The solution procedure then involves repeatedly solving this for each node in turn, hopefully converging to the true solution. The numerical equation for the local objective function for a node, i , where linear interpolation functions are used is

$$\Xi_i = \sum_j w_j \xi(\theta_i, \theta_j), \quad (18)$$

where j is for the nodes of the one element ring about node i and w_i is the combined areas of the elements sharing edge ij . Its unconstrained optimisation with respect to θ_i is executed by the BFGS solver in the Scipy library [49].

A numerical smoothing procedure for optimising a cross-field to improve its smoothness is outlined in Alg. 3. The crosses are smoothed incrementally, one and then the next, over the mesh.

Algorithm 3 Smooth cross-frame field

```

settings: numIterations,  $\Delta\xi_{\text{threshold}}$ 
activeNodes  $\leftarrow$  {all nodes (except those with fixed crosses)}
iteration  $\leftarrow$  1
while (iteration  $\leq$  numIterations and activeNodes  $\neq$  {}) do
  nextActiveNodes  $\leftarrow$  {}
  for node $i$  in activeNodes do
    Locally optimise cross $i$  ... (Minimise Eqn. (18) by BFGS solver)
     $\Delta\xi \leftarrow \xi(\theta_i, \theta_j)$ 
    if  $\Delta\xi > \Delta\xi_{\text{threshold}}$  then
      Add node $i$  and {node $j$ } to nextActiveNodes
  activeNodes  $\leftarrow$  nextActiveNodes
  iteration++

```

Each adjustment at a node is made according to the current values of neighbouring nodes. The order at which nodes are attended to is not critical. A pre-set number of iterations can be performed or it can be continued until the difference in the optimised angle in consecutive iterations falls below a threshold. The cost of the smoothing method is dramatically reduced after the first iteration

where local optimisation is applied to all nodes. In subsequent iterations local optimisation is only conducted at nodes where there was a significant change previously and at their neighbours. A significant change is deemed to be $\xi_{threshold} = 0.01$, which is an arbitrarily chosen figure that equates to roughly a 3° change in cross angle. This technique greatly accelerates the rate of convergence of the method without seeming to compromise its effectiveness.

An undesirable effect of smoothing is the undoing of the properties built into the cross-field in the initialisation stage for accommodating the target sizes and directions. It may be possible to add new terms to the smoothing function to counteract this although this has not been thoroughly explored. Thus, excessive smoothing is avoided and a maximum of three smoothing iterations is imposed as a matter of course.

The results of the smoothing method applied to the initialised cross-fields in Fig. 7 are shown in Fig. 9.

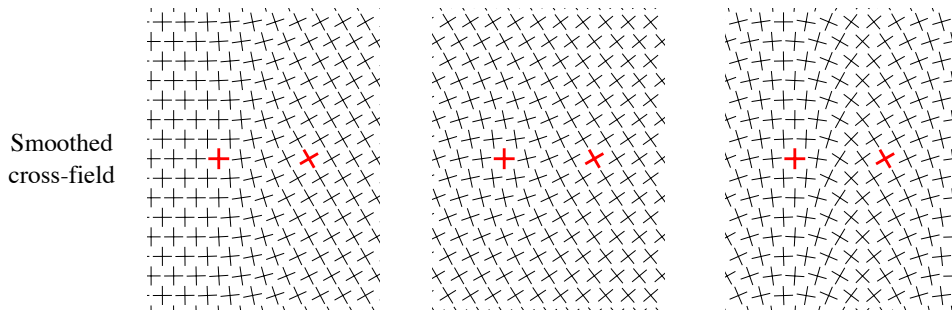


Figure 9. Improved cross-fields of Fig. 7 after three smoothing iterations.

4.8. Identification of singularities

Having found a cross for every node the singularities can then be extracted. Each edge is assigned the minimum change in angle, $\Delta\theta$, (in the interval $(-\pi/4, \pi/4]$) to move one cross at an end node to the other. If the surface normals are not co-aligned then one cross is rotated to the other's tangent plane to make the measurement, as for a discrete parallel transport operation. Singularities occur in tri elements if the cross-field cannot be continuously interpolated with the assigned $\Delta\theta$ s for its edges. For a tri element with nodes 1, 2 and 3, with conventional anti-clockwise ordering, it follows from Bunin's theory [19] that the index of the contained singularity is given by

$$k = -\frac{2}{\pi} \left(\Delta\theta_{12} + \Delta\theta_{23} + \Delta\theta_{31} + \iint_e K dA \right). \quad (19)$$

The total Gaussian curvature term can be neglected if the tri mesh is generated with curvature sensitivity and the differences of surface normals at adjacent nodes is not too great, and it is sufficient to round k to the nearest integer. Only simple singularities ($k = \pm 1$) will be found if minimum angle changes (in the range $[-\pi/4, \pi/4]$) are assigned to edges. Illustrations of the singularity identification method are shown in Fig. 10.

Unless there is an important targeted rapid change in mesh size it is usually undesirable for singularities of opposite types to occur in close proximity because they entail a high local distortion. Positive-negative singularity pairs occurring in adjacent tri elements can be simply removed by redefining the angle change for the shared edge by

$$\Delta\theta \rightarrow \Delta\theta - \text{sign}(\Delta\theta) \frac{\pi}{2}. \quad (20)$$

This technique is used by default in the implemented algorithm, although in practice smoothing eliminates these incidences automatically.

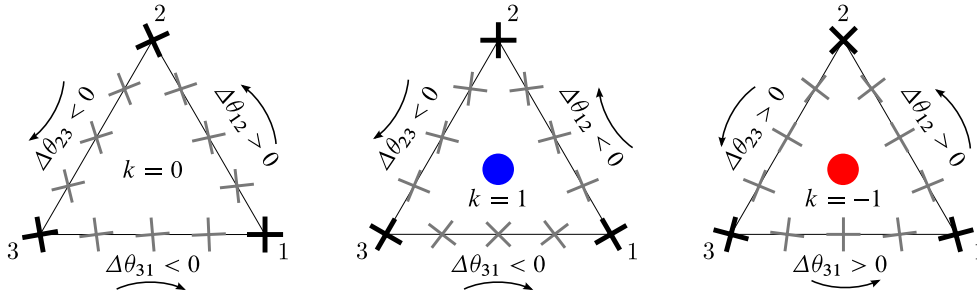


Figure 10. Singularity identification in tri elements from assigned changes in cross angle on the edges.

5. GENERATION OF MULTIBLOCK DECOMPOSITIONS

The cross-field describes the local orientations of quad elements and establishes the positions of mesh singularities. The next step is to decompose the surface into blocks inside which regular grids can be mapped with low distortion. The separatrices of the cross-field radiating from singularities and boundary corners have the effect of partitioning the surface into topological quad blocks. Therefore, tracing these over the tri mesh is the fundamental step in generating the multiblock decomposition.

5.1. Numerical streamline tracing

There are existing methods that are widely used for generating streamlines of (velocity) vector fields, $\mathbf{v}(\mathbf{x})$, by numerically integrating the equation

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x}). \quad (21)$$

To trace a streamline of a cross-field there is a simple extra step of ascertaining which of the four local cross vectors the initial velocity corresponds to before proceeding as usual. A method following that outlined in [50] is used to trace streamlines through tri elements. The algorithm uses an explicit 2-stage Runge-Kutta method. The surface is treated as being planar over a tri element and crosses at nodes are rotated to the element plane by a minimum angle rotation in the 3D space.

The Runge-Kutta method does not apply in elements with singularities because continuity in the cross-field is necessary. Therefore some basic work-around strategies are used. The element is replaced by three new elements with a node at its centroid for the singularity. Three new edges span between the original element nodes and the new node at its centroid. Their $\Delta\theta$ values are set as the smallest rotation angles to relate the crosses at the original nodes to the singularity *star* (described below) at the other end. To trace streamlines in these new elements Heun's method, as described by Kowalski *et al.* [30], is used for which vector interpolation is only necessary over the tri element edges. Illustrations of the procedure are seen in Fig. 11 (right) where the streamline is drawn in dotted red.

Each point of a cross-field has four assigned oriented directions except those at singularities which have k *star* (orientated) directions. They are evenly distributed around the singularity at intervals of $2\pi/(k+4)$, as proved by Bunin [26, App. A]. To find the correct angular alignment of the singularity *star* a ray along a cross-field direction from a position on an original edge is searched for which passes through the singularity node. The parametric position where this occurs is given by

$$t = \frac{\alpha_0 - \theta_0}{\Delta\theta - (\alpha_1 - \alpha_0)}. \quad (22)$$

with the symbolic definitions indicated in Fig. 11 (left). This finds one appropriate direction of the singularity and the rest are obtained by rotating it $(k+4)$ times by $2\pi/(k+4)$.

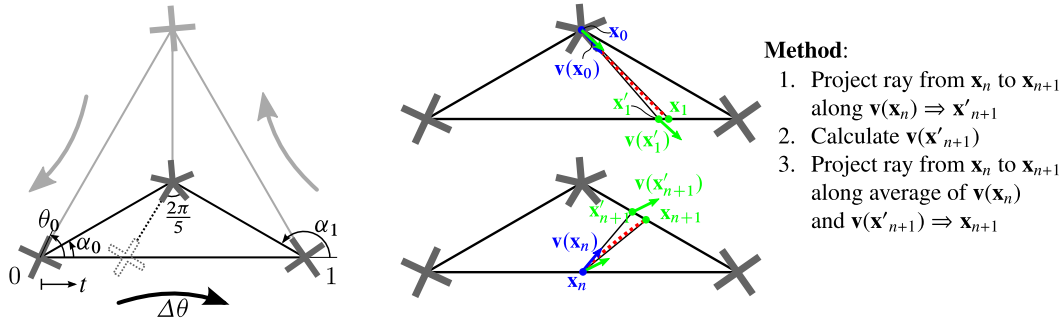


Figure 11. Orientating a singularity star (left) and numerical streamline tracing in singularity sub-element starting from singularity star direction (top right) and for an arbitrary streamline passing through (bottom right).

6. RESULTS

A series of results of the cross-field generation method and their traced separatrices will now be examined.

In Fig. 12 the results for a test surface, which is a simple planar surface with testing features, are seen for three different target metric tensor fields. The multiblock decompositions are pleasing to the eye which is a good first check for gauging their quality. The only topological feature which may draw disapproval is the polar grid topology in the corner on the lower right hand side of the geometry. A polar grid is associated with high order $k = -4$ type singularities (Fig. 2 (right)). Sometimes this type of mesh pattern is not permitted but to mesh this kind of osculating corner any other way would result in highly distorted elements. If a polar mesh is prohibited then then corner should be truncated beforehand. The first metric field is unit and uniform which is the default and it results in a simple block topology. The second and third metric fields are constant anisotropic and graded isotropic respectively and they cause the number and positions of the singularities to change so that the multiblock decompositions are more suited for realising the target element sizes. Values of 50 are used for both w_1 and w_2 in Eqn. (14). Increasing or decreasing their values changes the sensitivity of the method to size variation and anisotropy respectively, with the solutions becoming more similar to the first one as their assigned values approach zero.

In Fig. 13 the cross-field solution on a regular rectangle contains two negative and two positive singularities as a consequence of a high priority target grading of the mesh. The result is not perfectly symmetric due to numerical errors caused by the incremental extrapolation process over the irregular tri mesh in the initialisation method. A more refined tri mesh would result in a proportional increase in symmetry.

An example where additional fixed alignment constraints have been included is shown in Fig. 14. In order to accurately resolve the shock front in a CFD simulation it is imperative that the elements are aligned with the shock features to admit robust treatment in the solver [51] and also to make possible anisotropic grading without incurring severe skewing. The outlines of the 'lambda shocks' and a wake direction control feature were sketched roughly from the experimental flow image shown. These were included as *scratch* features of the surface, i.e. edges with two geometric orientations that essentially represent collapsed holes. These are treated as extra boundaries with the usual hard alignment constraints. *A posteriori* techniques have been developed for extracting these types of flow features from flow solutions and partially converged flow solutions [52, 53]. They can also be included *a priori* based on what is expected from acquired experience, as is done here, to induce a decomposition of an appropriate topology that permits effective r-type adaptation to the solution.

Two examples of the method applied to curved surfaces are shown in Fig. 15. Here the singularities emerge not as consequences of incompatible alignment constraints since the geodesic

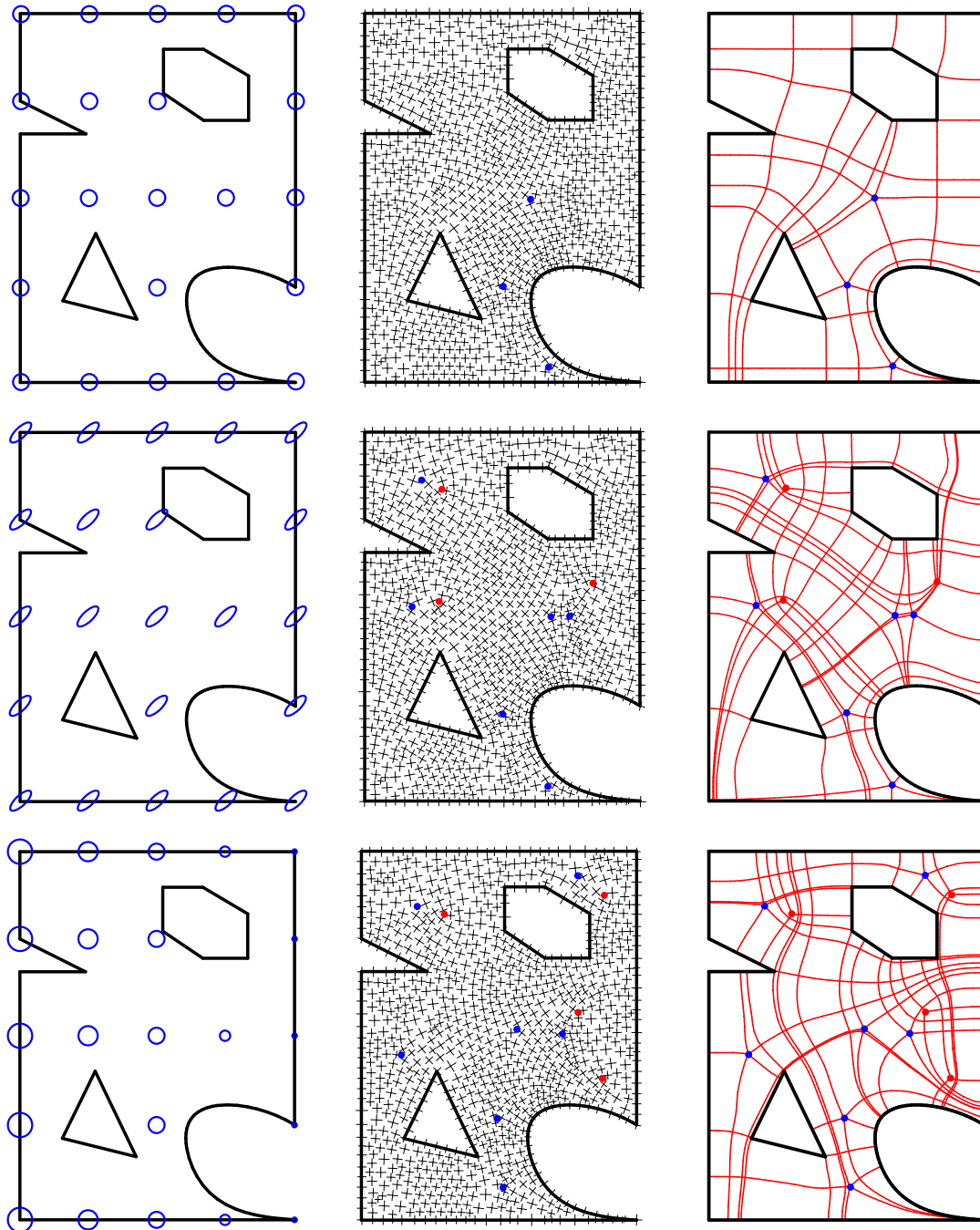


Figure 12. Results for a simple test surface with three different target metric tensor fields (rows). The target metric tensor field (left column), the cross-field solution (centre column) and the resulting multi-block decomposition by tracing all critical streamlines to the geometry boundaries (right column).

curvatures of all boundaries are zero and all corner angles are $\pi/2$. Rather, they are due to the Gaussian curvatures of the surfaces.

A perfectly symmetrical circular surface is seen in Fig. 16. The cross-fields produced after the initialisation phase and after the subsequent smoothing process with five iterations are shown left and right. The initialisation algorithm that is based on propagating information from the boundaries

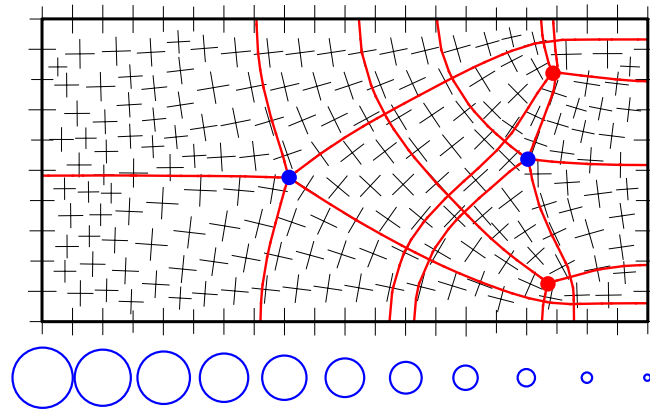


Figure 13. Result with high-priority metric tensor field on rectangular surface

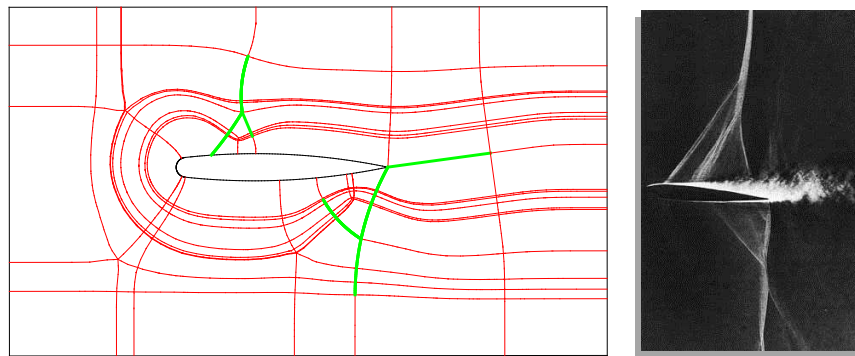
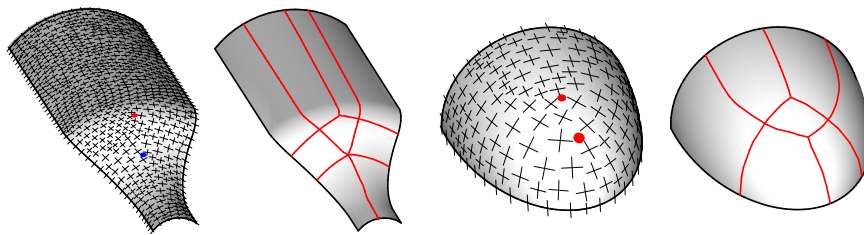


Figure 14. Result for simple aerofoil geometry with added alignment constraints (in green) representing shock fronts and wake in transonic flow solution, as seen in experimental flow image (taken from [54]).

Figure 15. Results for *quarter of wine bottle* and *quarter of ellipsoid* surfaces.

tends to create cross-fields with singularities on the medial axis. Thus, the singularities are clustered near the centre of the circle initially. Numerical error accounts for the singularities not all being superimposed exactly on the centre. Smoothing has the effect of repelling the singularities from each other and reorganising the cross-field.

A realistic 2-D CFD domain with a multi-element aerofoil in landing configuration is seen in Fig. 17. The cove region between the main body and flaps is sufficiently complex that working out the interfaces between curvilinear coordinate meshes fitted to each blade, as done by [55] for turbo machine geometries, is a big hurdle, particularly when the small hook feature is to be present in the mesh. The present method can effectively resolve the block structure in this region, as shown in the close-up view, and the critical streamlines form a well-shaped decomposition for the configuration, albeit a quite intricate and complex system of blocks.

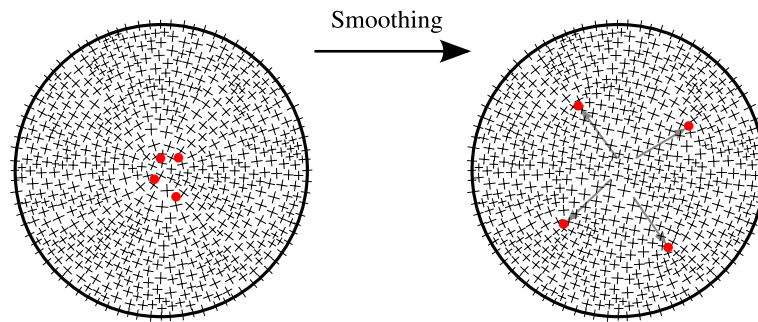


Figure 16. Cross-field results before and after smoothing (10 iterations) on a circular surface.

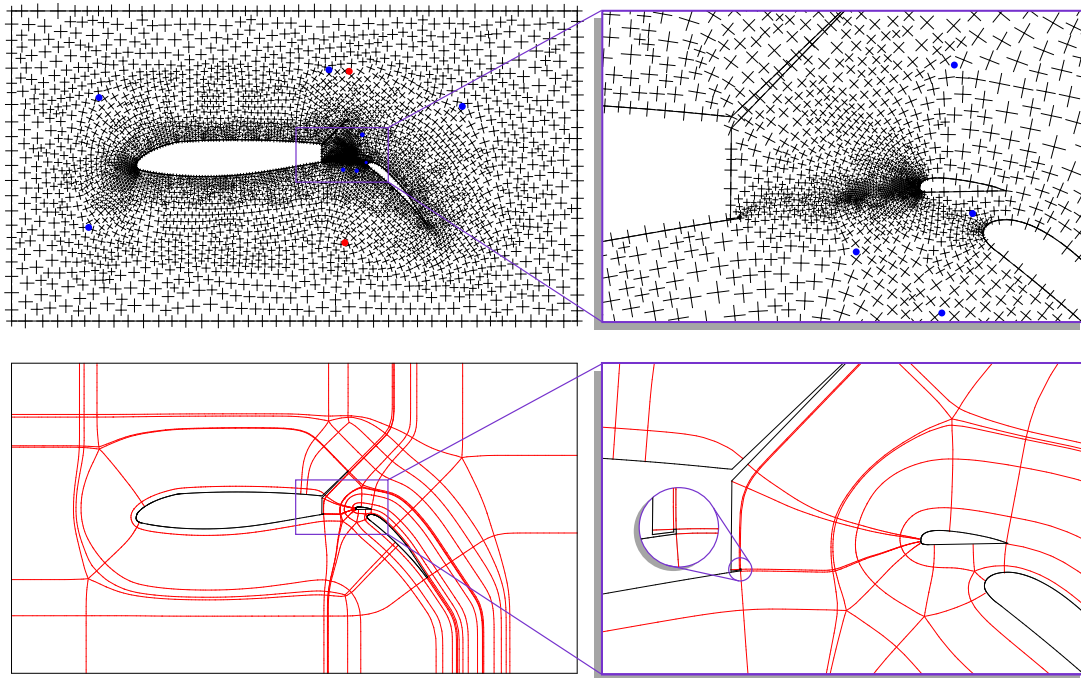


Figure 17. Result for ARA *multi-element aerofoil in landing configuration* geometry.

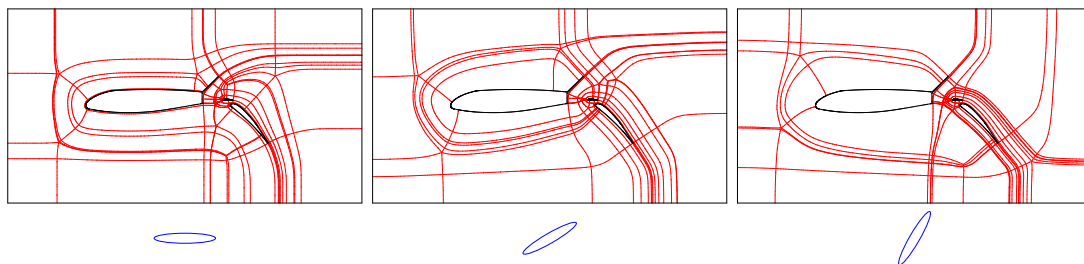


Figure 18. Result for ARA *multi-element aerofoil in landing configuration* geometry with target mesh anisotropy in different directions.

The results of the method re-run with constant anisotropic metric tensor fields with principal directions at an angles of 0 , $\pi/6$ and $\pi/3$ to the horizontal are shown in Fig. 18. Each has an improved suitability for the generation of a stretched mesh in the stream-wise direction with flows at different angles of attack. Even better results could be obtained by putting more work into tailoring a befitting metric tensor field using knowledge of the expected flows properties. For example, the

flow must travel smoothly around the aerofoils so in adjacent regions the metric tensors should have anisotropy in-line with the aerofoil contours. In the cove region the flow becomes stagnated so isotropic sizes are appropriate. Moving away from the aerofoils into the far-field, the metric tensors should gradually transition into the constant anisotropic metric tensor aligned with the main-stream flow. The combination of this practice of metric tensor field design along with the addition of fixed alignment constraints in important regions gives the user plenty of scope for controlling the shape and topology of the decomposition.

The example shown in Fig. 19 highlights a limitation of the proposed method. Although the cross-field solution superficially appears to be good, spiralling streamlines are seen to occur which do not allow a non-degenerate fitted quad mesh. One separatrix spirals indefinitely about the hole and tracing is terminated (marked by green arrow) after a maximum distance is exceeded.

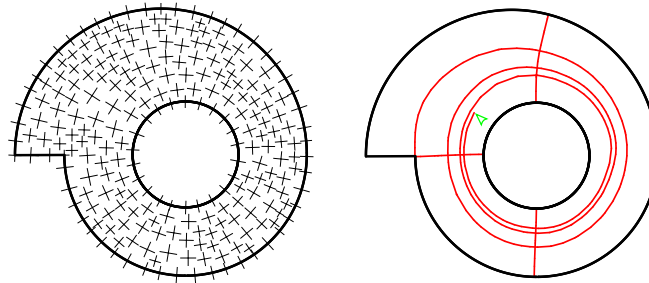


Figure 19. Simple example with a spiralling separatrix.

7. IDENTIFIED PROBLEMS AND PROPOSED SOLUTIONS

The previous examples exhibited the types of multiblock decompositions that are generated by the method. Overall, they have good topologies and shape qualities but there are evidently some issues that need addressing before they can be practically useful. These are:

1. There are excessive numbers of blocks.
2. There can be very narrow blocks.
3. The separatrices may not terminate at boundaries but can have a free end where tracing is stopped to avoid spiralling.
4. A series of attached blocks can have a large variation in height, potentially entailing a large disagreement between actual and target mesh sizes.

The first issue is due to the separatrix graph being more complex than is necessary for a multiblock decomposition. The use of 'super blocks' or merged adjacent blocks that together form topological rectangles and allowing partial contiguity between blocks would significantly simplify the decompositions.

The second issue is caused by separatrices passing by each other at distances much smaller than the target element size. Ideally, they should form a single block edge. Often this is caused by two separatrices passing each other at a small distance when they should actually join each other in order to satisfy harmony between the symmetry of the surface and symmetry of the solution. The sources of these problems are numerical errors in the initialisation of the cross-field and integration of streamlines and the imprecise location of singularities.

The third issue pertains to spiralling streamlines. The only way out of this problem is to insert an additional positive-negative singularity pair to increase the topological flexibility and cause some streamlines to be redirected. (More on this after Fig. 21.)

The fourth issue arises because of the limited way in which element sizes are dealt with. Although the method makes adjustments to the orientations of crosses to facilitate target size variation, in

practice these are little more than ‘nudges in the right direction’ and the influence of the alignment constraints tends to dominate. Therefore, the final mesh topology might force mesh sizes to deviate considerably from the target sizes. The use of positive-negative singularity pairs in a small number of well-chosen blocks is often needed to achieve acceptable agreement with the target sizes. These are called *transition primitives* [56, Sec.17.7.5] and are commonly used when interactively meshing block decompositions.

More serviceable multiblock decompositions can be generated by adding a simple termination rule to the separatrix tracing method where separatrices cut-off each other’s paths, as shown in Figs. 20. There are significantly fewer blocks in the resulting multiblock decompositions that now

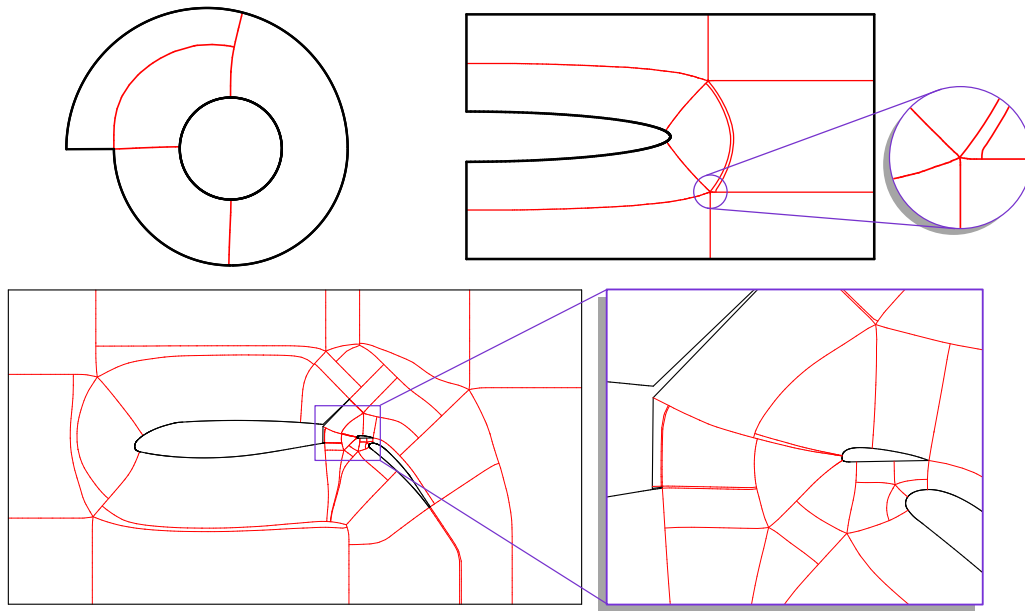


Figure 20. Multiblock decompositions with partial block contiguity

have partial block contiguity. The issue of indefinite spiralling of streamlines also seems to be prevented provided that every hole has at least one corner of any type. In particular, a smooth hole should be treated as having at least one corner with an angle of π . A proof of this assertion has not been constructed so its truth is only speculated; it is supported by empirical testing and a counter example is elusive. This modification adequately deals with the first and third issues of the list.

To generate a quad mesh on the decomposition appropriate integer division numbers on the block edges must be found that preferably match the target sizes closely. The target division numbers are computed by estimating the integral of the target size field along the block edges, dividing by the edge length and rounding to the nearest integer. A number of methods have been developed to solve this problem [35, 36, 37, 38]. Partial block contiguity of the block decompositions makes finding the optimum solution significantly more difficult but finding a feasible solution of adequate optimality is usually still comfortably attainable. The most important step in the methods is the assignment of suitable *mesh primitives* to the blocks. The default template is the $m \times n$ grid and it demands equality constraints in the division numbers across opposite sides of a block, as outlined in Fig. 21. With the mesh primitives assigned the problem can be effectively solved by minimising a simple objective function involving the differences between the edge divisions and their targets (provided that the constraints do not have inconsistencies). An integer linear programming formulation of this was described first by Tam and Armstrong [35]. More recently, Mitchell [38] described a sophisticated solver with a piecewise linear objective function that can evenly distribute the discrepancies from the target sizes.

The constraints given in Fig. 21 have an inconsistency in that they are only satisfied if $N_1 = N_3 = 0$. This can be identified by the methodology of [57] where zero eigen-values are picked up on in

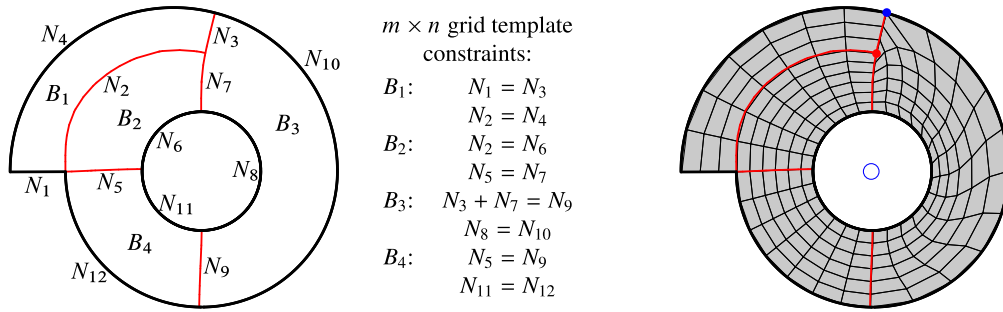


Figure 21. Division number constraints for $m \times n$ grid templates on multiblock decomposition (left) and mesh for which one transition template has been used (right).

the constraint equations. But since N_1 relates to the division number on a boundary a zero division number is not permitted. Thus, one $m \times n$ grid template must be switched for a transition primitive. A mesh where block B_3 has been assigned a transition primitive is shown in Fig. 21 (right) where the positive and negative singularities are highlighted by blue and red dots. This was actually solved automatically in CADfix.

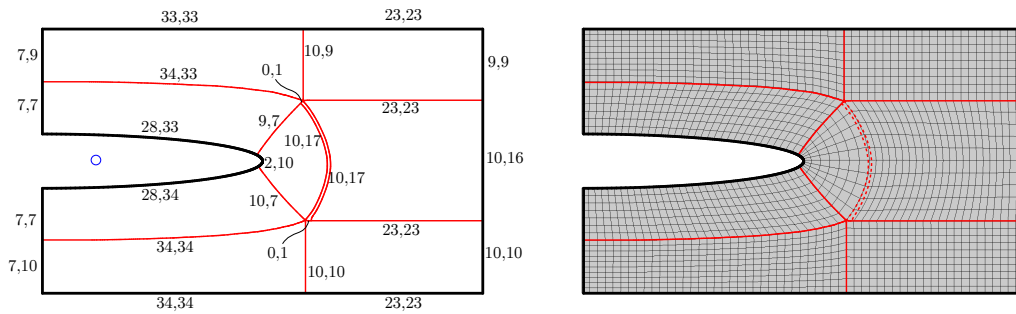


Figure 22. Division numbers (target, solved) and corresponding block-structured meshes on rounded trailing edge surface.

In Fig. 22 (left) target and solved division numbers for the multiblock decomposition of the rounded trailing edge surface with a global target size indicated by the small blue circle. The division numbers are solved using the ‘division balancing’ function in CADfix [46]. The division numbers are constrained to be at least one which forces those on the very small edges near the two positive singularities up from the optimal values of zero. The block-structured mesh seen in Fig. 22 (right) has been created on a manually altered multiblock decomposition to produce the ideal result where zero division numbers are permitted and the dotted edges bounding the very thin block appears as a single edge in the mesh. New meshing methods that can automatically achieve this would be extremely useful. They should be able to deal with zero division numbers and would use the multiblock decomposition as a flexible framework and not as an immutable structure. In such a way the second issue could be got around.

The mesh shown in Fig. 23 was generated in Abaqus [56] with considerable manual tinkering of the division numbers and trial-and-error selection of transition primitives.

An easy way to improve the shape qualities of the elements in the meshes shown in Figs. 21–23 would be to apply smoothing methods to the whole mesh without block edge conformity constraints (using Mesquite Mesh Quality Improvement Toolkit [58] for example). In particular, the almost degenerate element seen at the negative singularity in Fig. 21 (right) would be remedied by simple node repositioning algorithms.

Simultaneously optimising division numbers while finding appropriate mesh templates for blocks is a tough problem. Mohring *et al.* [36] formulate an exact combinatorial problem to describe this

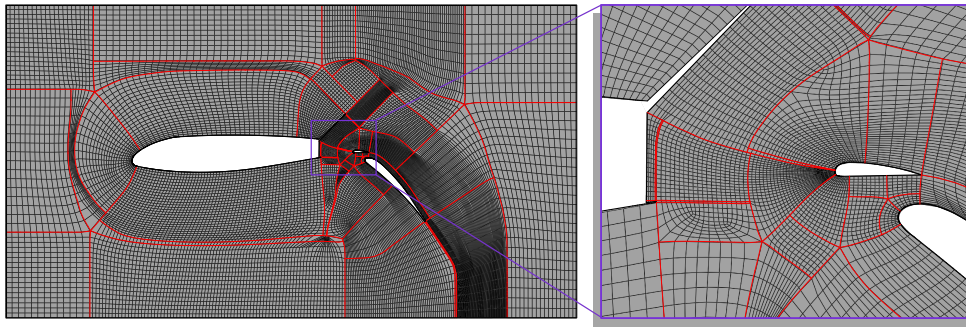


Figure 23. Block-structured mesh on *multi-element aerofoil in landing configuration geometry*.

and show that it is NP-complete when discrepancy bounds are enforced. Despite this shadow of intractability they developed a heuristic technique based on the idea of network flows that can achieve pretty good results reliably for instances of real-life complexity. These methods could be applied directly to the multiblock decompositions of the present method.

8. CONCLUSIONS

A method has been presented that can effectively generate a cross-field on surfaces with boundaries. By using an efficient *fast marching method* to organise the order in which new crosses are computed a consistent orderly outcome is ensured, that being, smoothly varying cross-fields everywhere except across the medial axis of the surface. The method is controllable in the sense that it makes adjustments according to a metric tensor field and additional alignment constraints can be easily handled.

A method is also described for automatically generating a multiblock decomposition from a cross-field. At first look, the separatrices of the cross-field appear to provide effective partitioning edges. However, the complete separatrix graph was found to be inadequate due to excessive numbers of blocks, very narrow blocks and spiralling behaviour. To overcome these setbacks a multiblock decomposition with partial continuity that is composed of separatrix segments was proposed. Generating a quad mesh on the multiblock decomposition involves solving integer division numbers and choosing mesh templates for the blocks. Existing methods can effectively carry this out this task although new customisations are needed in order to handle zero division numbers which are often necessary to achieve the best block-structured meshes. This will be an area of future work.

ACKNOWLEDGEMENT

This work was sponsored by the Aircraft Research Association (ARA). The authors would like to thank Transcendata for their support.

APPENDIX A

In this appendix the formula is derived for finding the optimum angle of a free cross at a node node in a tri element with respect to the fixed crosses at the other two nodes, taking into consideration the smoothness and target gradient and directions. The set up is shown in Fig. 24 (left) where the crosses at nodes 1 and 2 are fixed and the cross at node 3 is free.

A.1 Linear tri element theory

In a tri element with nodes 1,2,3 (ordered anti-clockwise with respect to the element positive normal) that have local 2-D Cartesian coordinates $\mathbf{x}_1 = (x_1, y_1)$, $\mathbf{x}_2 = (x_2, y_2)$, $\mathbf{x}_3 = (x_3, y_3)$ the *Barycentric*

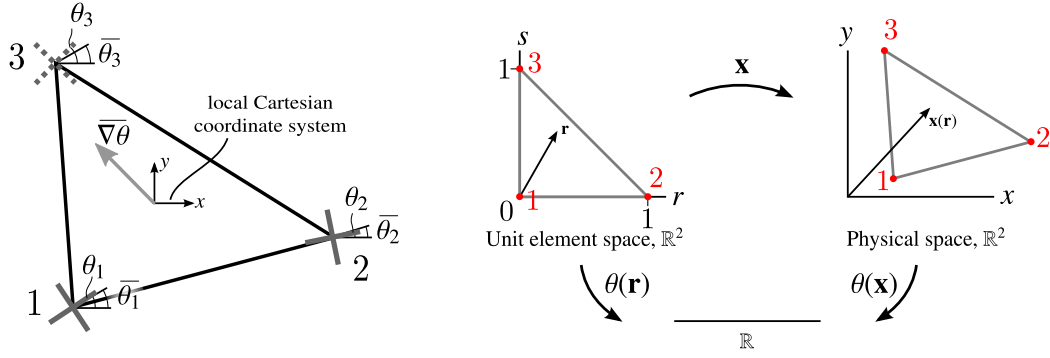


Figure 24. The arrangement of the problem (left) and a diagram of the transformation from unit element space to physical space for a linear tri element and their respective mappings of θ to the real number line (right).

coordinates, (L_1, L_2, L_3) have the properties

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} \quad (23)$$

$$\Rightarrow \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (24)$$

If L_1, L_2 and L_3 are substituted for $(1 - r - s), r$ and s respectively, an affine map from unit element space, $\mathbf{r} = (r, s)$, to physical space, $\mathbf{x} = (x, y)$, is obtained by rearranging Eqn. (23):

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \Rightarrow \mathbf{x} = \mathbf{J}\mathbf{r} + \mathbf{x}_1. \quad (25)$$

\mathbf{J} is the constant Jacobian or differential matrix, $\begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial s} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial s} \end{bmatrix}$, of the transformation $\mathbf{x}(\mathbf{r})$. From Eqn. 24, the inverse of the transform is given by

$$\begin{bmatrix} r \\ s \end{bmatrix} = \underbrace{\begin{bmatrix} a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}}_{\mathbf{J}^{-1}} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c_2 \\ c_3 \end{bmatrix}. \quad (26)$$

The Barycentric coordinates of the linear tri element can also be used as linear shape functions to interpolate the unknown (the cross angle, θ , in this case) over the element:

$$\theta(\mathbf{x}) = [\theta_1 \quad \theta_2 \quad \theta_3] \begin{bmatrix} L_1(\mathbf{x}) \\ L_2(\mathbf{x}) \\ L_3(\mathbf{x}) \end{bmatrix}. \quad (27)$$

Substituting again for r and s , θ in terms of \mathbf{r} is given by

$$\theta(\mathbf{r}) = [\theta_2 - \theta_1 \quad \theta_3 - \theta_1] \begin{bmatrix} r \\ s \end{bmatrix} + \theta_1 \quad (28)$$

for which the differential matrix is

$$\mathbf{d}\theta(\mathbf{r}) = [-\nabla\theta(\mathbf{r})] = \begin{bmatrix} \frac{\partial \theta}{\partial r} & \frac{\partial \theta}{\partial s} \end{bmatrix} = [\theta_2 - \theta_1 \quad \theta_3 - \theta_1]. \quad (29)$$

By the chain rule the differential matrices with unit element coordinates and physical coordinates are related by

$$\begin{aligned}\mathbf{d}\theta(\mathbf{r}) &= \mathbf{d}\theta(\mathbf{x}) \circ \mathbf{J}, \\ \mathbf{d}\theta(\mathbf{x}) &= \mathbf{d}\theta(\mathbf{r}) \circ \mathbf{J}^{-1},\end{aligned}\tag{30}$$

which supplies a convenient expression for determining the gradient of θ in physical space using Eqn. 29 and the coefficients in Eqn. (24),

$$\nabla\theta(\mathbf{x})^\top = [\theta_2 - \theta_1 \quad \theta_3 - \theta_1] \begin{bmatrix} a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}.\tag{31}$$

A.2 Derivation of formula

The optimum value for θ_3 is sought and the input variables are: $\theta_1, \theta_2, \bar{\theta}_1, \bar{\theta}_2, \bar{\nabla}\theta, w_1, w_2$ and the geometric attributes of the tri element. Referring to Eqn. (14), the optimum value for θ_3 satisfies

$$\frac{\partial E}{\partial \theta_3} = \frac{\partial E_{\text{smoo}}}{\partial \theta_3} + w_1 \frac{\partial E_{\text{grad}}}{\partial \theta_3} + w_2 \frac{\partial E_{\text{direc}}}{\partial \theta_3} = 0.\tag{32}$$

Using Eqn. (31),

$$\begin{aligned}E_{\text{smoo}}^{(e)} &= \frac{1}{2} \int_e \|\nabla\theta(\mathbf{x})\|^2 dA \\ &= \frac{1}{2} \|\nabla\theta^{(e)}(\mathbf{x})\|^2 A^{(e)} \\ &= \frac{A^{(e)}}{2} [\theta_2 - \theta_1 \quad \theta_3 - \theta_1] \begin{bmatrix} a_2 & b_2 \\ a_3 & b_3 \end{bmatrix} \begin{bmatrix} a_2 & a_3 \\ b_2 & b_3 \end{bmatrix} \begin{bmatrix} \theta_2 - \theta_1 \\ \theta_3 - \theta_1 \end{bmatrix}\end{aligned}\tag{33}$$

where the superscript (e) indicates a property belonging to the tri element. Then,

$$\begin{aligned}\frac{\partial E_{\text{smoo}}^{(e)}}{\partial \theta_3} &= \frac{A^{(e)}}{2} \left([0 \quad 1] \begin{bmatrix} a_2 & b_2 \\ a_3 & b_3 \end{bmatrix} \begin{bmatrix} a_2 & a_3 \\ b_2 & b_3 \end{bmatrix} \begin{bmatrix} \theta_2 - \theta_1 \\ \theta_3 - \theta_1 \end{bmatrix} \right. \\ &\quad \left. + [\theta_2 - \theta_1 \quad \theta_3 - \theta_1] \begin{bmatrix} a_2 & b_2 \\ a_3 & b_3 \end{bmatrix} \begin{bmatrix} a_2 & a_3 \\ b_2 & b_3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \\ &= A^{(e)} \left((a_2 a_3 + b_2 b_3)(\theta_2 - \theta_1) + (a_3^2 + b_3^2)(\theta_3 - \theta_1) \right)\end{aligned}\tag{34}$$

And for the gradient term,

$$\begin{aligned}E_{\text{grad}}^{(e)} &= \int_e \|\nabla\theta - \bar{\nabla}\theta\|^2 dA = \left\| \begin{bmatrix} a_2 & a_3 \\ b_2 & b_3 \end{bmatrix} \begin{bmatrix} \theta_2 - \theta_1 \\ \theta_3 - \theta_1 \end{bmatrix} - \begin{bmatrix} g_x \\ g_y \end{bmatrix} \right\|^2 A^{(e)} \\ &= \left((a_2(\theta_2 - \theta_1) + a_3(\theta_3 - \theta_1) - g_x)^2 + (b_2(\theta_2 - \theta_1) + b_3(\theta_3 - \theta_1) - g_y)^2 \right)\end{aligned}\tag{35}$$

where $\bar{\nabla}\theta^{(e)} = \text{const} = (g_x, g_y)$. Differentiating and rearranging produces

$$\frac{\partial E_{\text{grad}}^{(e)}}{\partial \theta_3} = 2A^{(e)} \left((\theta_2 - \theta_1)(a_2 a_3 + b_2 b_3) + (\theta_3 - \theta_1)(a_3^2 + b_3^2) - (a_3 g_x + b_3 g_y) \right).\tag{36}$$

Similarly for the direction term,

$$\begin{aligned}E_{\text{direc}}^{(e)} &= \frac{1}{A^{(e)}} \int_e (\theta - \bar{\theta})^2 dA \\ &= \frac{1}{A^{(e)}} \iint_e \left(\theta^{(e)}(\mathbf{x}) - \bar{\theta}^{(e)}(\mathbf{x}) \right)^2 dx dy \\ &= \frac{1}{A^{(e)}} \iint_e \left(\theta^{(e)}(\mathbf{r}) - \bar{\theta}^{(e)}(\mathbf{r}) \right)^2 |\mathbf{J}| ds dt.\end{aligned}\tag{37}$$

Since $|\mathbf{J}| = 2A^{(e)}$,

$$E_{\text{direc}}^{(e)} = 2 \iint_e \left(\theta^{(e)}(\mathbf{r}) - \overline{\theta^{(e)}}(\mathbf{r}) \right)^2 ds dt. \quad (38)$$

Both $\theta^{(e)}(\mathbf{r})$ and $\overline{\theta^{(e)}}(\mathbf{r})$ are linear polynomial functions over the element so their difference squared is a quadratic function and a three-point quadrature rule can be used calculate the integral without any loss in accuracy [59, Ch. 10.4]. Applying this whilst using Eqn. (28) for $\theta^{(e)}(\mathbf{r})$ and similarly for $\overline{\theta^{(e)}}(\mathbf{r})$ gives

$$\begin{aligned} E_{\text{direc}}^{(e)} = 2 \left(\frac{1}{6} \left([\theta_2 - \theta_1 \quad \theta_3 - \theta_1] \begin{bmatrix} \frac{1}{6} \\ \frac{1}{6} \end{bmatrix} + \theta_1 - [\bar{\theta}_2 - \bar{\theta}_1 \quad \bar{\theta}_3 - \bar{\theta}_1] \begin{bmatrix} \frac{1}{6} \\ \frac{1}{6} \end{bmatrix} - \bar{\theta}_1 \right)^2 \right. \\ + \frac{1}{6} \left([\theta_2 - \theta_1 \quad \theta_3 - \theta_1] \begin{bmatrix} \frac{2}{3} \\ \frac{1}{6} \end{bmatrix} + \theta_1 - [\bar{\theta}_2 - \bar{\theta}_1 \quad \bar{\theta}_3 - \bar{\theta}_1] \begin{bmatrix} \frac{2}{3} \\ \frac{1}{6} \end{bmatrix} - \bar{\theta}_1 \right)^2 \\ \left. + \frac{1}{6} \left([\theta_2 - \theta_1 \quad \theta_3 - \theta_1] \begin{bmatrix} \frac{1}{6} \\ \frac{2}{3} \end{bmatrix} + \theta_1 - [\bar{\theta}_2 - \bar{\theta}_1 \quad \bar{\theta}_3 - \bar{\theta}_1] \begin{bmatrix} \frac{1}{6} \\ \frac{2}{3} \end{bmatrix} - \bar{\theta}_1 \right)^2 \right). \quad (39) \end{aligned}$$

Differentiating and simplifying yields

$$\frac{\partial E_{\text{direc}}^{(e)}}{\partial \theta_3} = \frac{\theta_1}{6} + \frac{\theta_2}{6} + \frac{\theta_3}{3} - \frac{\bar{\theta}_1}{6} - \frac{\bar{\theta}_2}{6} - \frac{\bar{\theta}_3}{3}. \quad (40)$$

Bringing together Eqns (34), (36) and (40) in Eqn. (32) and simplifying using the properties $a_1 + a_2 + a_3 = 0$ and $b_1 + b_2 + b_3 = 0$ the sought formula for the optimum angle of the cross at node 3 is achieved:

$$\begin{aligned} \theta_3 = \frac{1}{(1 + 2w_1)A^{(e)}(a_3^2 + b_3^2) + \frac{w_2}{6}} \left(\theta_1 \left(- (1 + 2w_1)A^{(e)}(a_1a_3 + b_1b_3) - \frac{w_2}{6} \right) \right. \\ + \theta_2 \left(- (1 + 2w_1)A^{(e)}(a_2a_3 + b_2b_3) - \frac{w_2}{6} \right) \\ \left. + \left(2w_1A^{(e)}(a_3g_x + b_3g_y) - \frac{w_2}{6}(\theta_{\text{tar}1} + \theta_{\text{tar}2} + 2\theta_{\text{tar}3}) \right) \right). \quad (41) \end{aligned}$$

REFERENCES

1. Hallquist J. *LS-DYNA3D Theoretical Manual*. Livermore Software Technology Corporation 1991. <http://www2.nsysu.edu.tw/csmlab/fem/dyna3d/theory.pdf> (Accessed: 11/2013).
2. Jensen M, Du Bois P, Ho P. LS-Dyna analysis for structural mechanics 2011. http://www.predictiveengineering.com/Solutions/products/training/docs/ls-dyna/LS-DYNA_Analysis_for_Structural_Mechanics-Partial_Note_Set.pdf (Accessed: 11/2013).
3. Chen L, Shen J, Xu C. A unstructured nodal spectral-element method for the Navier-Stokes equations. *Commun. Comput. Phys.* 2012; **12**(1):315–336.
4. Komatitsch D, Tsuboi S, Tromp J. *The Spectral-Element Method in Seismology*. American Geophysical Union, 2013; 205–227, doi:10.1029/157GM13.
5. Aftosmis M, Gaitonde D, Tavares TS. Behavior of linear reconstruction techniques on unstructured meshes. *AIAA Journal* 1995; **33**:2038–2049, doi:10.2514/3.12945.
6. Ait-Ali-Yahia D, Baruzzi G, Habashi WG, Fortin M, Dompierre J, Vallet MG. Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part II: Structured grids. *International Journal for Numerical Methods in Fluids* 2002; **39**(8):657–673, doi:10.1002/flid.356.
7. Rumsey CL. HiLiftPW-2 Summary. *2nd AIAA CFD High Lift Prediction Workshop*. 2013. <http://hiliftpw.larc.nasa.gov/Workshop2/ParticipantTalks/HLPW2-rumsey.pdf> (Accessed: 3/2014).
8. Blacker TD, Stephenson MB. Paving: A new approach to automated quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering* 1991; **32**(4):811–847, doi:10.1002/nme.1620320410.
9. Staten ML, Kerr RA, Owen SJ, Blacker TD, Stupazzini M, Shimada K. Unconstrained plastering–hexahedral mesh generation via advancing-front geometry decomposition. *International Journal for Numerical Methods in Engineering* 2010; **81**(2):135–171, doi:10.1002/nme.2679.
10. Schneiders R. A grid-based algorithm for the generation of hexahedral element meshes. *Engineering with Computers* 1996; **12**(3–4):168–177, doi:10.1007/BF01198732.
11. White D. Automatic quadrilateral and hexahedral meshing of pseudo-cartesian geometries using virtual subdivision. Master's Thesis, Brigham Young University 1996.

12. Tam TKH, Armstrong CG. 2-D finite element mesh generation by medial axis subdivision. *Advances in Engineering Software* 1991; **13**(5/6):313–324.
13. Rigby D. TopMaker: A technique for automatic multi-block topology generation using the medial axis. *Technical Report NASA/CR-213044* 2004.
14. Hertzmann A, Zorin D. Illustrating smooth surfaces. *Proceedings of SIGGRAPH 2000*, 2000; 517–526.
15. Wei LY, Levoy M. Texture synthesis over arbitrary manifold surfaces. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, ACM: New York, NY, USA, 2001; 355–360, doi:10.1145/383259.383298.
16. Alliez P, Cohen-Steiner D, Devillers O, Lévy B, Desbrun M. Anisotropic polygonal remeshing. *ACM Trans. Graph.* Jul 2003; **22**(3):485–493, doi:10.1145/882262.882296.
17. Palacios J, Zhang E. Rotational symmetry field design on surfaces. *ACM Trans. Graph.* Jul 2007; **26**(3), doi:10.1145/1276377.1276446.
18. Ray N, Vallet B, Li WC, Lévy B. N-symmetry direction field design. *ACM Transactions on Graphics*, 2008. Presented at SIGGRAPH.
19. Bunin G. A continuum theory for unstructured mesh generation in two dimensions. *Comput. Aided Geom. Des.* Jan 2008; **25**(1):14–40.
20. Hon Y, Li M, Melnikov Y. Inverse source identification by green's function. *Engineering Analysis with Boundary Elements* 2010; **34**(4):352 – 358, doi:10.1016/j.enganabound.2009.09.009.
21. Shewchuk JR. What is a good linear element? - Interpolation, conditioning, and quality measures. In *11th International Meshing Roundtable*, 2002; 115–126.
22. Gray A, Abbena E, Salamon S. Ch. 19. Principle curves and umbilic points. *Modern Differential Geometry of Curves and Surfaces with Mathematica, Third Edition*. Chapman & Hall/CRC, 2006.
23. Marinov M, Kobbelt L. Direct anisotropic quad-dominant remeshing. *Proceedings of the 12th Pacific Conference on Computer Graphics and Applications*, 2004; 207 – 216, doi:10.1109/PCCGA.2004.1348351.
24. Kälberer F, Nieser M, Polthier K. Quadcover - surface parameterization using branched coverings. *Computer Graphics Forum* 2007; **26**(3):375–384, doi:10.1111/j.1467-8659.2007.01060.x.
25. Ben-Chen M, Gotsman C, Bunin G. Conformal flattening by curvature prescription and metric scaling. *Computer Graphics Forum* 2008; **27**(2):449–458.
26. Bunin G. Towards unstructured mesh generation using the inverse poisson problem 2008. *arXiv:0802.2399*.
27. Hildebrandt K, Polthier K, Wardetzky M. Smooth feature lines on surface meshes. *Proceedings of the Third Eurographics Symposium on Geometry Processing*, SGP '05, Eurographics Association: Aire-la-Ville, Switzerland, Switzerland, 2005. URL <http://dl.acm.org/citation.cfm?id=1281920.1281935>.
28. Pouget M, Cazals F. Approximation of ridges and umbilics on triangulated surface meshes. *CGAL User and Reference Manual*. 4.3 edn., CGAL Editorial Board, 2013. URL http://doc.cgal.org/4.3/Manual/packages.html#PkgRidges_3Summary.
29. Bommes D, Zimmer H, Kobbelt L. Mixed-integer quadrangulation. *ACM Trans. Graph.* Jul 2009; **28**(3):77:1–77:10, doi:10.1145/1531326.1531383.
30. Kowalski N, Ledoux F, Frey P. A pde based approach to multidomain partitioning and quadrilateral meshing. *Proceedings of the 21st International Meshing Roundtable*, Jiao X, Weill JC (eds.). Springer Berlin Heidelberg, 2013; 137–154.
31. Liu Y, Xu W, Wang J, Zhu L, Guo B, Chen F, Wang G. General planar quadrilateral mesh design using conjugate direction field. *ACM Trans. Graph.* Dec 2011; **30**(6):140:1–140:10, doi:10.1145/2070781.2024174.
32. Bommes D, Campen M, Ebke HC, Alliez P, Kobbelt L. Integer-grid maps for reliable quad meshing. *ACM Trans. Graph.* Jul 2013; **32**(4):98:1–98:12, doi:10.1145/2461912.2462014.
33. Bommes D, Lempfer T, Kobbelt L. Global structure optimization of quadrilateral meshes. *Computer Graphics Forum* 2011; **30**(2):375–384, doi:10.1111/j.1467-8659.2011.01868.x.
34. Li Y, Wang W, Ling R, Tu C. Shape optimization of quad mesh elements. *Comput. Graph.* Jun 2011; **35**(3):444–451, doi:10.1016/j.cag.2011.03.037.
35. Tam TKH, Armstrong CG. Finite element mesh control by integer programming. *International Journal for Numerical Methods in Engineering* 1993; **36**(15):2581–2605, doi:10.1002/nme.1620361506.
36. Möhring RH, Müller-Hannemann M, Wiehe K. Mesh refinement via bidirected flows: Modeling, complexity, and computational results. *J. ACM* May 1997; **44**(3):395–426, doi:10.1145/258128.258174.
37. Mitchell S. High fidelity interval assignment. *Int. J. Comput. Geometry Appl.* 2000; **10**(4):399–415.
38. Mitchell S. Simple and fast interval assignment using nonlinear and piecewise linear objectives. *Proceedings of the 22nd International Meshing Roundtable*, Sarrate J, Staten M (eds.). Springer International Publishing, 2014; 203–221, doi:10.1007/978-3-319-02335-9_12.
39. Sethian JA. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. 2 edn., Cambridge University Press, 1999.
40. do Carmo MP. *Riemannian Geometry*. 1st ed. 1992. corr. 14th printing 2013 edn., Birkhäuser, 2011.
41. Frey PJ, George PL. Ch. 10. Quadratic forms and metrics. *Mesh Generation*. Wiley-ISTE, 2010; 331–360, doi:10.1002/9780470611166.ch10.
42. Alauzet F. Metric-based anisotropic mesh adaptation June 2010. Numerical Analysis Course, http://www-roc.inria.fr/gamma/Frederic.Alauzet/cours/cea2010_V2.pdf (Accessed: 25th August 2013).
43. Tchon KF, Khachan M, Guibault F, Camarero R. Three-dimensional anisotropic geometric metrics based on local domain curvature and thickness. *Computer-Aided Design* 2005; **37**(2):173 – 187, doi:10.1016/j.cad.2004.05.007.
44. Vyas V, Shimada K. Tensor-guided hex-dominant mesh generation with targeted all-hex regions. *Proceedings of the 18th International Meshing Roundtable*, Clark B (ed.). Springer Berlin Heidelberg, 2009; 377–396, doi:10.1007/978-3-642-04319-2_22.
45. Makem JE, Armstrong CG, Robinson TT. Automatic decomposition and efficient semi-structured meshing of complex solids. *Engineering with Computers* 2012; :1–17doi:10.1007/s00366-012-0302-x.

46. ITI Transcendata. CADfix website. <http://www.transcendata.com/products/cadfix> (Accessed: 02/2013).
47. Elias RN, Martins MAD, Coutinho ALGA. Simple finite element-based computation of distance functions in unstructured grids. *International Journal for Numerical Methods in Engineering* 2007; **72**(9):1095–1110, doi: 10.1002/nme.2079.
48. Python Software Foundation. 8.4. heapq – Heap queue algorithm, Python Language Reference, version 2.5. <http://www.python.org>.
49. Jones E, Oliphant T, Peterson P, *et al.*. SciPy: Open source scientific tools for Python. <http://www.scipy.org/> 2001–.
50. Dorobantu M. Efficient streamline computations on unstructured grids. *Technical Report TRITA-NA-9709*, Royal Institute of Technology (KTH) Sept 1997.
51. Carpenter MH, Casper JH. Accuracy of shock capturing in two spatial dimensions. *AIAA Journal* 1999; **37**(9):1072–1079.
52. Qin N, Liu X. Flow feature aligned grid adaptation. *International Journal for Numerical Methods in Engineering* 2006; **67**(6):787–814, doi:10.1002/nme.1648.
53. Harris M. Flow feature aligned mesh generation and adaptation. PhD Thesis, University of Sheffield, Department of Mechanical Engineering 2013.
54. Baals DD, Corliss WR. Wind tunnels of NASA. *Technical Report*, NASA, Scientific and Technical Information Branch 1981. Online: <http://history.nasa.gov/SP-440/cover.htm> (Accessed 18/12/13).
55. Shahpar S, Lapworth L. Padram: Parametric design and rapid meshing system for turbomachinery optimisation. *ASME Turbo Expo 2003, collocated with the 2003 International Joint Power Generation Conference*, GT2003-38698, 2003; 579–590.
56. SIMULIA Dassault Systemes. Abaqus 6.11 User's Manual.
57. Spekrijse SP, Boerstoele JW. An algorithm to check the topological validity of multi-block domain decompositions. *Numerical Grid Generation in Computational Field Simulations*, NLR, International Society of Grid Generation, 1998; 161–170.
58. Brewer ML, Diachin LF, Knupp PM, Leurent T, Melander DJ. The mesquite mesh quality improvement toolkit. *IMR*, 2003. URL <http://dblp.uni-trier.de/db/conf/imr/imr2003.html#BrewerDKLM03>.
59. Akin JE. *Finite Element Analysis with Error Estimators: An Introduction to the FEM and Adaptive Error Analysis for Engineering Students*. 1 edn., Butterworth-Heinemann, 2005.